TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

ATTE LAMMINSALO

VISUALIZATION PIPELINE FOR LOCATION-BASED DATA

Master's thesis

Examiners: Assistant professor Davide Taibi and Professor Kari Systä

The examiners and topic of the thesis were approved on 28 November 2018

ABSTRACT

ATTE LAMMINSALO: Visualization pipeline for location-based data
Tampere University of Technology
Master of Science Thesis, 44 pages
November 2018
Master's Degree Programme in Science and Engineering
Major: Software Engineering
Examiners: Assistant professor Davide Taibi and Professor Kari Systä

Keywords: location-based data, visualization, pipeline, audience-oriented approach

The master's thesis focused on answering how to turn a plethora of location-based data into a visualization pipeline. The goal was to find a way to bring the most value to the users of the visualization pipeline, in this case a taxi company, by defining a set of audience-oriented approaches. The audience-oriented approaches were tested using a case study, using the taxi company as the test subject. The case study was used to test the functionality and design of the visualization pipeline, as well as to test the audience-oriented approaches in practice. The success of the visualization pipeline was assessed using a Business Intelligence assessment model. The Business Intelligence assessment was used as a benchmark for how much value the implemented visualization pipeline provided.

We were able to define three types of audiences, and correspondingly three different approaches for these said audience types. The three audience types were activists, analysts and organizational decision-makers, and their approaches were defined correspondingly as ”lightweight”, ”technical”, and ”tailored” approaches.

The case study was carried out by defining an audience group for the customer. The case study defined the case customer as an organizational decision-maker, thus resulting the ”tailored” approach as the best fit. The approach provided the most value to the case customer both in the terms of technical requirements as well as in the terms of data analytical needs. This case study showed promise for the utility of an audience-oriented approach in visualization pipeline design.

TIIVISTELMÄ

ATTE LAMMINSALO: Sijaintipohjaisen datan visualisointiputki
Tampereen teknillinen yliopisto
Diplomityö, 44 sivua
Marraskuu 2018
Tieto- ja sähkötekniikan diplomi-insinöörin tutkinto-ohjelma
Pääaine: Ohjelmistotuotanto
Tarkastajat: Apulaisprofessori Davide Taibi ja Professori Kari Systä

Avainsanat: sijaintidata, visualisaatio, visualisaatioputki, yleisö-orientoitu lähestymistapa

Diplomityössä vastattiin kysymykseen miten suuri määrä sijaintipohjaista dataa voidaan muuntaa visualisointiputkeksi. Tavoitteena oli löytää keino tuottaa mahdollisimman paljon arvoa visualisointiputken käyttäjille määrittelemällä yleisö-orientoituja lähestymistapoja. Käyttäjinä olivat tässä tapauksessa taksiyrityksen ylempi johtoporras. Yleisö-orientoiduille lähestymistavoille tehtiin tapaustutkimus, jossa taksiyritys oli tapauskohde. Tapaustutkimuksen tarkoituksena oli todentaa visualisointiputken toiminallisuus yleisellä tasolla, sekä testata yleisö-orientoitua lähestymistapaa käytännön tasolla. Visualisointiputken tuomaa arvoa todennettiin myös liiketoimintatiedon hallintamallin -arviointimallilla.

Diplomityössä määriteltiin kolmenlaiset yleisötyypit, ja vastaavasti lähestymistavat kullekin yleisötyypille. Nämä kolme yleisötyyppiä olivat aktivistit, analyytikot ja organisatoriset päätöksentekijät. Näitä vastaavat lähestymistavat olivat ”kevyt”lähestymistapa, ”tekninen”lähestymistapa ja ”räätälöity”lähestymistapa.

Tapaustutkimus suoritettiin ensin määrittelemällä tapauskohteelle yleisötyyppi. Tapauskohde määräytyi organisatoriseksi päätöksentekijäksi, ja täten sile määräytyi ”räätälöity”lähestymistapa. Lähestymistapa tuotti eniten arvoa tapauskohteelle teknisesti sekä dataanalyyttisesti. Tapaustutkimus myös osoitti lupausta yleisö-orientoidulle lähestymistavalle visualisointiputkien suunnittelussa.

PREFACE

First, I would like to thank my wonderful master's thesis examiner and supervisor, Davide Taibi. Without your help and guidance, my master's thesis would have been an utter mess. I also want to thank everyone at the university for making the process as smooth as it was. Also, a special shout out to Mari Torikka for being the best Academic Officer on the planet, without you I would not have been able to graduate in only four and a half years.

I would also like to thank the marvellous bunch at Innogiant Oy. You let me help create value to your clients with extraordinary generosity. I truly feel my research made a difference, which means a lot to me. I want to especially thank Markus Rentto, Mika Vuori and Osmo Someroja for being my direct support at the company and for letting me bounce my ideas around without a moment of hesitation or frustration. It meant a lot to me, really. You will always have a special place in my heart.

I want to thank my lovely fiancée Anni, who I will have the honor to marry this coming March. You have been my cornerstone for all these years and hopefully for the years to come as well. You are the most hard working person I know, and it inspired me to push my own limits. I love you. I also love our latin dance hobby and how we will have even more time for it now.

Throughout these years I have been active in teekkari culture. I want to thank Skilta, my home at school. Even though I changed my major, I never abandoned my electric brothers and sisters, nor did they abandon me. I also got the honor of serving as Skilta's chairman of the board. Thank you especially to the board of 2017, but also to the other boards I got to serve in. Thank you to Shark Council, you rock. I also want to thank TTYY, where I was a Wappu Secretary. It was one of the most fun things I have ever experienced, especially thanks to my fellow Golden Boys. I want to thank my friends at BFI, you were there from the start, and you will stay with me to the bitter end, I know it. I want to thank Elram, because some of my most fond memories from university were made with you guys. I also want to thank the chat group Dänk Memes, because not only did you help me push through my master's thesis, you inspired me to be the meme lord I am today.

In Tampere, Finland, on the 3rd of December 2018

Atte Lamminsalo

CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

HTTP	Hypertext Transfer Protocol
MVC	Model-view-controller
HTML	Hypertext Markup Language
PDF	Portable Document Format
JSON	JavaScript Object Notation
API	Application Program Interface
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
REST	Representational State Transfer

1. INTRODUCTION

”It is a capital mistake to theorize before one has data”, a quote made famous by Sherlock Holmes in Arthur Conan Doyle’s famous A Scandal in Bohemia [5]. A quote from over 100 years ago, still valid today. Only, Sherlock Holmes did not speak of theorizing over business data, but that is beyond the point. In today’s world we are supplied with data that is waiting to be stored and analyzed, and this is where the master thesis at hand comes into play. The predicament where the master’s thesis starts in is tightly knit with the need of data visualization, which in turn lets us theorize and analyze.

This chapter is written to bring the reader up to speed on the work description and problem definition, as well as the possible assumption and limitations. The chapter gives background for the reasons the research is done.

1.1 Problem definition

It is often problematic when you have a set of raw data, which you want to refine into a clear visual representation. In this ever-changing world of Internet of Things, data is often at large. Turning this data into information, on which you could base your business decisions on is not as straightforward as one might think. It gets even more problematic when the data is location-based, because this often requires an implementation of a map visualization. Maps have a tendency to be hard to analyse if implemented incorrectly.

This thesis aims to answer the question many software engineering consultants ask themselves when working with a client with a need for data refinement – how to turn a multitude of raw location-based data into a clear visual representation, and what are the different approaches to the problem to find the best one for maximum value for the client at hand?

The main goal of the master’s thesis work was to provide generalized approaches to creating a visualization pipeline for different audience types. The goal was to be able to provide a starting point for developers designing visualization pipelines with an audience-oriented approach, and to test the newly defined approaches using a case study. The more imminent goal was to provide value to Innogiant Oy’s customer by using them as a case customer to test the visualization pipeline designed in this master’s thesis.

A multitude of different tools exist for data analysis and visualization, but it is unclear on how well these tools work together with different frameworks, especially when it comes to location-based data. It is also unclear how different audience types can benefit from different types of frameworks used in the visualization of the data. The frameworks can vary from whole data analysis platforms to simple web frameworks.

1.2 Work description

The thesis work was done for a company called Innogiant Oy, and was done in cooperation with their client. The client company is a taxi company from the Tampere area, and has provided Innogiant Oy a third-party API to access locations of specific taxis, their ride state, and a plethora of other data. Although the data endpoint was provided easily, the data itself was in shambles. A lot of the data was repeated, analytically uninteresting, or in the worst case, inexplicable. On top of this, the endpoint was not documented, which caused a lot of problems in development.

The case customer had an existing business intelligence dashboard application where the thesis work would later be embedded into. At this time, the data volumes of the case customer were not in the scale of big data, but it was nonetheless something that needed to be taken into account in regards to future proofing the pipeline.

The taxi reform that took place in July of 2018, has been a driving force for data analysis tools in the taxi industry, and a big motivator for the work done in this master's thesis.

1.2.1 Taxi reform

Since July 1st 2018, Finland's previously heavily regulated taxi industry took a turn to a more liberal market [22]. Until the reform, there was a previously determined maximum tariff which in most cases was set as the standard running price for taxis. The reform included change in fares and pricing, change in how one can obtain a taxi license as well as a reform in how one can order a taxi. The reform allows for taxi companies to adopt new service models to improve their services. Authorities expected the change to cause 90% of future taxi orders to be done via mobile applications. [22]

The visualization pipeline aims to visualize how practices carried over from times prior to the reform are inefficient, e.g. the notoriously long taxi queues next to taxi poles.

1.3 Work requirements

The basic requirements of the visualization pipeline for location-based data was based on the needs of the client company, which will be used as a test case later in this master's thesis to evaluate its success. The very basic necessity for analytics comes from the need of data to make business decisions, i.e. to lower customer wait times or to lower customer cancellations, thus increasing customer satisfaction and profit margins.

In the case of the case customer, the visualization pipeline would help visualize locations where taxis are idling instead of picking up rides. The technical requirements for the pipeline are not as strict, and the only significant requirements were the capability to embed or couple the solution into their current dashboard solution and future proofing the pipeline for later upgrade. The technical limitations are gone through in section 1.4.

1.4 Assumptions and limitations

Some assumptions and limitations have been taken into account during the thesis work. Mostly the assumptions and limitations focus on the existing infrastructure with the case customer. The aforementioned infrastructure is explained in further detail in later chapters. Concisely, the implemented data visualization pipeline started from a taxi location API provided by a third party vendor, which in turn limited some of our possibilities. Assumptions on the workings of the API have to be made, due to the lacking documentation of the API.

To make the thesis easier to follow, the following assumptions and limitations are good to know:

- The taxi location API was from a third party vendor, thus was out of the scope of the thesis research question.
- The API data was taken as is and was trusted to be accurate enough for the thesis.

1.5 Methodology

The research of the thesis is qualitative by nature, and this is why a case study was the chosen method for the research in this master's thesis. The research strategy was to research different audience-oriented approaches, implement these approaches and test an approach with the case study. The testing was done using by assessing the provided value to the case customer from a data analytical point of view, as well as a technical point of view. Innogiant Oy's customer was used as the case subject in the master's thesis.

The visualization pipeline design was also assessed using a Business Intelligence assessment model with the case study. The function of the Business Intelligence assessment model was to assess the overall success of the pipeline.

1.6 Thesis outline

The thesis is split into two parts. In the first section the basis for the data visualization pipeline is researched, in addition to defining the different audience types for data visualizations. The theory section consists of the following steps:

- Researching data pipelines.
- Defining the requirements for a modern implementation of the pipeline.
- Familiarizing the common concepts of web applications.
- Defining the requirements for a modern web application.
- Researching data visualization.
- Defining the audience types for data visualizations.

- Researching data analysis and business development.

In the second part the theory is tested by implementing the visualization pipeline for the case customer, based on the research. The main focus of the research was how to categorize different audience types and how the requirements for visualizations vary with these audiences to be able to make generalizations. Business development was studied in addition to the methods for assessing how well an implemented Business Intelligence can provide value to an organization.

The technical focus of the research done in the first section was set on the front end of the pipeline, and how to build a proper web application. The back end was researched to be able to design it in such a way that it could be later upgraded with minimal effort. On top of this, visualization was researched in general to create the best possible visualizations for the case customer.

The second part, the implementation section, consists of the following steps:

- Selecting the visualization types.
- Filtering the visualization tools based on their functionalities.
- Filtering the visualization tools further based on the different audiences.
- Choosing the programming languages used.
- Defining the evaluation method for the case study.
- Implementing a modern pipeline for location-based data visualization.
- Documenting the pipeline.
- Documenting the problems faced with the pipeline implementation.
- Evaluating the case customer audience type.
- Comparing the different approaches and visualization tools to the case customer.
- Choosing the correct approach for the case customer.
- Evaluating the success of visualization pipeline for the case customer.

This part of the thesis focuses primarily on the case customer, and how different approaches in the front end of the visualization pipeline can affect the end result, in addition to documenting the implemented visualization pipeline. The goal was to find the right approach for the case customer, thus testing the generalized approaches researched in the first section. The research question was not to compare which visualization tool was superior, because it was clear that every tool that passes the primary filtering was going to visualize the data. It can also be argued that comparing the tools alone does not bring research value in the long term, because the tools get updated and new tools are expected to emerge.

The thesis is split further into six different chapters. The first chapter covers the background and the basics of the research, such as the problem definition, work description, work requirements, assumptions and limitations, methodology and the thesis outline.

Chapter 2 is the theoretical background, covering all the theory implemented and used in further chapters. This chapter defines many of the important angles the research takes, such as the visualization audience and the assessment method later used in chapter 5 to evaluate the success of the visualization pipeline.

Whereas chapter 3 covers the methodologies of the study in further depth. This includes the technology choices of both the back end and front end of the pipeline, as well as the chart types used in the visualizations. In this chapter the filter process of the visualization tools is gone in depth. The different approaches are designed based on the different audience types studied in 2. The designs include the visualization tools which are used in the different approaches.

Next, in chapter 4, the implementation of the visualization pipeline for location-based data is covered, both as a whole data flow architecture, as well as in depth with each component involved. The problems faced in the implementation are documented in this chapter.

Chapter 5 focuses on documenting the findings. The generalization of the different approaches are evaluated by using the case customer. This is done by categorizing the case customer, and by choosing the best approach. The pipeline as a whole is evaluated with the assessment method defined in chapter 2 by using the case customer as an example. A technical comparison is briefly visited in this section to find out whether the tools are interchangeable within the methods.

Finally, chapter 6 concludes the master's thesis and gives a final analysis of the work, as well as explores the possibilities of future work in the scope of this thesis.

2. BACKGROUND

This chapter is the basis for the research done in this thesis. The goal of the chapter is to study different aspects of the problem at hand to form a solid understanding of the scope of the problem, as well as to gain valuable knowledge on the possible ways to go on to solve the problem.

In this chapter, we first go through technical knowledge required in the implementation, i.e. data pipelines in general and the technologies utilized. After that, we examine the best ways to visualize location-based data, and define the audience types for each visualization approach. Finally, we conclude the chapter by studying how we can utilize visualizations to develop businesses, and what are the ways to assess the maturity of the business analytics.

2.1 Data pipelines

It is fundamental to have a competent strategy for capturing data as it is produced. The most typical and well tested approach for a data pipelines is to transform data within the pipeline. The disadvantage of this approach is that it is complex due to the fact that the transformation logic is kept in the data pipeline. If the data pipeline is replaced, the transformation logic needs to be reimplemented. On top of this the raw data is lost, and the authenticity of the data is lost. [21]

In addition to this data pipeline approach two other approaches are proposed in the Journal of Big Data [21]. The second approach is to transform data within the storage layer, which in turn brings in the advantage of making it easier to replace the current pipeline without having to reimplement the transform logic. This however requires larger storage space, as both the raw data and transformed data have to be stored and adds complexity to the storage layer with the transformation jobs. [21]

The third data pipeline approach is to transform the data in the analysis phase. This approach is simple in terms of the pipeline and makes it easy to replace. It requires less storage than the second approach, due to only having to store the raw data. The disadvantages are increased execution times in the visualization and analytics side of the pipeline, and a lot of repetition is needed, because the data needs to be transformed each time the analytics jobs are executed. Performance tests have shown that moving data transformation into the analytics job reduces overall processing time. [21]

2.2 Data models

JavaScript Object Notation, or JSON for short, is a lightweight format for data-interchange. The benefits of using JSON are plenty, namely it is easy for humans to read, while retaining

```
[
    {
        "id":"6083f96e-6483-42af-a23c-a4533c4d0f44",
        "creation_time":"2018-11-05T15:32:45.195Z",
        "car_id":"7923eca5-2f13-4522-b0c8-6f65d766d3fe",
        "snapshot_id":"acff644b-314a-4f07-ac52-76fa9820f884",
        "location":{
            "lat":61.50,
            "long":23.82
        },
        "speed":0,
        "heading":0,
        "ride_state":"DRIVING",
        "timestamp":"2018-11-05T15:32:33.000Z"
    }
]
```

Program 2.1. *Example of a JSON object.*

the property of being parseable for machines. As well as being totally language independent, JSON utilizes conventions that are familiar in most of the programming languages that exist, such as arrays and value/key pairs, making it optimal for data-interchange. [7] The data model can be seen in program 2.1

GeoJSON is a data model for geospatial data. It is based on JSON, and it defines several JSON objects into a combination defining geographic features, their properties and their spatial extents. The coordinate system uses the World Geodetic System 1984 standard with units of decimal degrees. [6] The format is visualized in program 2.2.

GeoJSON provides an advantage due to its uniform syntax. Visualization tools can take advantage this, by offering functionality that would be otherwise impossible with non-standardized data models.

2.3 Representational State Transfer

Representational State Transfer, or REST for short, is a style of web architecture, that is used to define quality attributes of the Web. It is identified as a loosely coupled, open, decentralized and massively distributed hypermedia system. The REST architectural design has design constraints to ensure it retains its intended qualities such as scalability, high performance and flexibility to name a few. [15]

The uniform interface is what is defined to be what sets RESTful APIs apart from other architectural styles. All of the web resources are identified by unique identifiers such as URIs, uniform resource identifiers. These identifiers are meaningful in the sense, that they can be represented outside of any context. The resources are conceptual, and the resource states can be retrieved and manipulated through requests. The uniform interface constraint requires for the state messages to be self-descriptive. The way the client and server interact

```
{
    "type":"FeatureCollection",
    "features":[{
        "type":"Feature",
        "geometry":{
                "type":"Point",
                "coordinates":[22.91,61.32]
        },
        "properties":{
                "id":"da73bff1-304e-4f6c-b02e-eaf64ab03681",
                "creation_time":"2018-11-05T15:45:15.196Z",
                "taxi_number":102,
                "vehicle_attributes":"28",
                "snap_shot_creation_time":"2018-11-05T15:45:15.155Z",
                "speed":44,
                "heading":17140,
                "ride_state":"DRIVING",
                "timestamp":"2018-11-05T15:45:09.000Z"
        }
    }]
}
```

Program 2.2. *Example of a GeoJSON object.*

is through exchange of requests and responses, which both include the data itself as well as the metadata concerning the request or response itself. [15]

A set of predetermined methods are defined for REST. In the example of the HTTP (Hypertext Transfer Protocol) protocol the list consists of the following:

- GET
- PUT
- DELETE
- POST
- HEAD
- OPTIONS

These all have well defined semantics on how they alter the state of the resource in question. The uniform interface has a final constraint, which is called the hypermedia constraint, which is simply a resource discovery mechanism. The idea is that the clients can discover resource identifiers through hyperlinks from related resources, and thus are able to navigate through the graph built from these relations. [15]

2.4 Web application architecture

Web application architectures that follow an MVC (Model-view-controller) design pattern, generally consist of five different conceptual layers. These layers are the domain layer,

user interface layer, the web layer, the service layer and data access layer. These layers are connected together and form the whole of a web application. The layers are connected one-sidedly, meaning the service layer can access the data access layer, but not vice versa. This design is implemented to make sure the architecture does not get too complex. These layers are visualized in figure 2.1. [4]

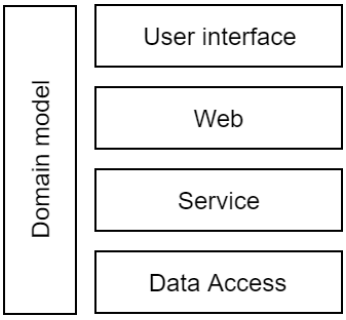

Figure 2.1. *Web MVC application layers adapted from [4].*

The domain layer is considered the most important layer of the whole architecture, because it represents the very essence of the problem the web application is trying to solve. This layer includes functionality such as checking whether an account can perform certain actions or not. [4]

The user interface layer is what the end user sees of the application. In a web browser the user would most likely see a HTML (Hypertext Markup Language) document or another client could request a PDF (Portable Document Format) file. [4]

The web layer has two roles, the first being responsible for navigating the user through the web application. The second is to integrate the service layer with HTTP. The first role can be as simple as routing the user through the website by mapping URLs. The second role handles HTTP requests and converts them into requests the service layer understands as well as transform the possible response from the server into something to display in the user interface. The web layer should be as thin as possible and should not have any business logic, because that is the service layer's role. [4]

The service layer controls the business logic, and is thought of as the core of the whole application. It exposes an API to the web layer. [4]

The data access layer accesses whatever storage system the web application uses. It does it in such a fashion that the service layer does not know which type of storage system is used. [4]

2.4.1 Remote rendering

Remote rendering is the practice of rendering on one computing device and displaying the results on another. This is usually done with applications that require a lot of calculating

power. The device that displays the results is usually referred as the client, while the device that does the computing is referred as the rendering server. The client can often contain a possibility for a user to interact with the data. The architecture basic architecture concept can be seen in figure 2.2. [20]

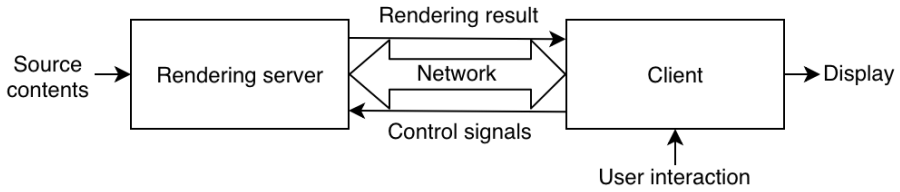

Figure 2.2. *Remote rendering server architecture adapted from [20].*

The advantages of remote rendering compared to the conventional approach of rendering locally is that it can provide weaker hardware, such as mobile devices, with faster results. On top of this computing power can be shared with multiple clients and repetition of the same calculations is not necessary. Remote rendering is also a good cross-platform solution, because the clients only need to be compatible with the rendering server. In some applications remote rendering can also prevent application contents leaking to malicious users, because only the rendering results are passed from the rendering server.

2.5 Big data

Even though the master's thesis data is not in the scale of big data, we briefly research big data to make sure we can future proof the pipeline. In big data, the three dimensions of big data are referred to as the 3Vs. These three Vs are volume, velocity and variety. [12]

Volume is the sheer amount of data that is created, which is usually in the petabyte scale in big data applications. Velocity is the speed in which the data is gathered, which usually is more important to take into account in real-time applications. Variety is the different types of data types processed, which often causes the problem of what data is actually relevant to gather. [12]

2.6 Data visualization

Data visualization is defined as a visual representation of data, that is created to convey the meaning and significance of data [16] [19]. Charts and graphs are used to understand the features and characteristics of the data, as well as used to summarize the data making it easier to grasp. [19]

Data visualizations allow us to not only show numerical data, but also to help visualize the changes in the data, as well as the shape of the data. The importance of charts and graphs are emphasized when the data sets are large, because this way you can visualize the data at a glance, instead of having to study it piece by piece. [19]

Less is more is usually the case when it comes to data visualization in general. But the problem with trying to simplify visualizations is finding the correct balance between simple and informative visuals.

2.6.1 Audience types

In data visualization it is important to be aware of the audience you wish to visualize the data to. Each audience comes from a different background and it is important to take this into account when creating any data visualization. [16]

For organizational decision makers, who are technically less inclined, it is often good to create simple visualizations, where the value can be extracted without much effort. It is important to tell a story with the data, because this way the audience can make more knowledgeable business decisions. Data storytelling is described to be a way to make sense of the data outside the visualization to bring context to the visualization. These visualizations are often static images, but interactive dashboards are also a good option for technically more proficient decision makers. [16]

For analysts it is often smarter to create interactive data visualizations. This way the viewer can explore the data themselves and form own opinions. The data visualization can be open-ended and thus promote the exploration of the data. In this kind of visualization, it is important to give as much context to the data as possible, so the analysts can draw conclusions from the given data. These visualizations can be interactive dashboards or static images for example. [16]

For activists, such as idealists or change-makers, data visualization is often used to make a point. In these cases the visualization should leave no room for interpretation and instead make a statement, that is easy to grasp onto. This kind of audience requires data art, which is in its very essence meant to entertain or provoke the audience to make a statement. [16]

Nonetheless, it is important to remember in data visualization and research in general that it is not ethical to distort data. All representations of data should be accurate and should not be altered just to please a certain audience. This is especially the case when it comes to non-technical audiences, who blindly trust the visualizations, because they lack the knowledge to recognize falsely represented data. [16]

Audience-oriented design is defined as designing something with the audience in mind. It considers the audiences values, biases and preconceptions. It takes into account language preferences such as professional jargon. It requires the designer to establish a connection with the audience, by viewing the design from their shoes. [1]

2.6.2 Colors

A big part of data visualization is the colors that are used to represent data. In general, it is quite natural to have different colors to represent different data types, but in terms of clarity

colors should have some implication for what they represent.

A powerful tool for selecting the colors for visualizing data on maps called Color Brewer was created at the GeoVISTA Center at Penn State University, which was later rebuilt online in 2009 [3]. The tool lets the user specify the data type and gives different options for colors. The different options for data schemes are sequential, qualitative and diverging. Sequential schemes are optimized for data that ranges from high to low, diverging schemes place equal emphasis on mid-range values as extremes, and qualitative schemes are good for categorical data [18]. The tool also informs if the selected colors are color-blind friendly or suitable for photocopiers among other things.

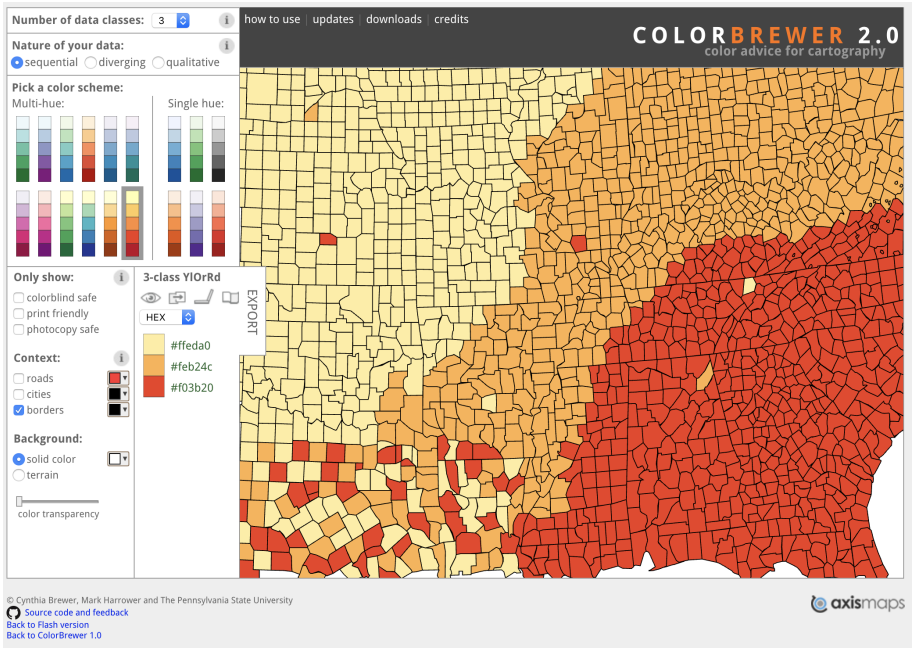

Figure 2.3. *Screenshot of the ColorBrewer 2.0 tool [3].*

2.6.3 Location-based data visualization types

Coordinate data can be visualized in multiple ways, such as markers, clusters or as a heatmap. Markers are the most straightforward approach to visualize point data on a map, simply add a dot or marker on the map where the coordinates point to. Point clusters are a combination of nearby markers, shown as a single marker, which shows how many points are close to each other either by showing a number or growing in size. These bring us to different types of location-based data visualization charts.

Heatmaps are basically brush strokes on the map on the specified coordinates, which change into another color or change in intensity, based on the amount of points in the vicinity. They are a good way to make points with certain features pop on a map. For example, if a heatmap shows meteor impact sites around the world, based on the crater size the brushed area could be made bigger. This makes it easier for the viewer to find the sites with big craters. Another good example is a carbon monoxide map, where areas range from different

colors based on the amount of carbon dioxide in the atmosphere. This can be seen in figure 2.4.

It is often difficult to define the perfect colors for a heatmap, let alone multiple heatmap layers on the same map representing different features. In cases of multiple heatmap layers, usually it is a good idea to make the most important data visually the most distinct. The different visual variables are position and different color gradients which can represent different properties.

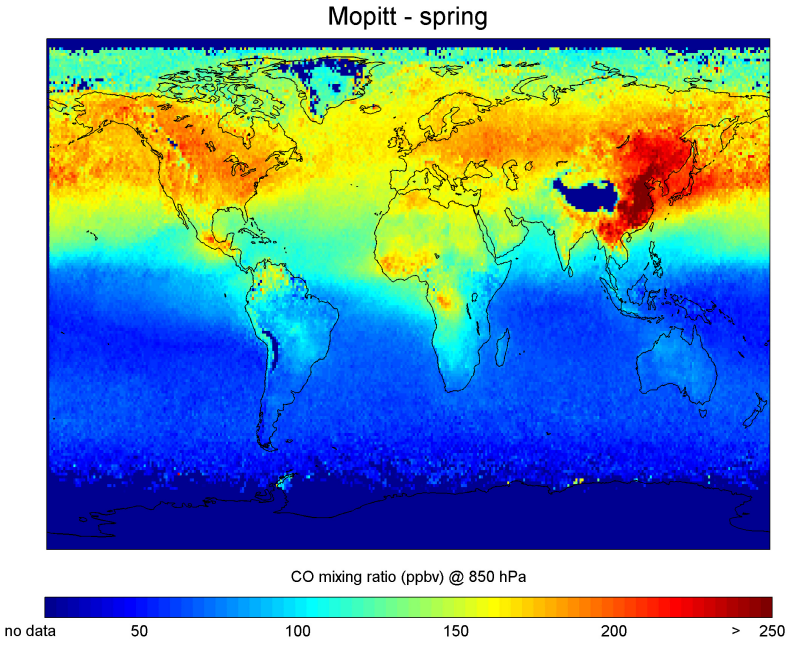

Figure 2.4. *Heatmap visualization of carbon monoxide concentrations in spring from [2]*

Another great choice for visualizing the relationships between multiple variables are scatter plot matrices [16]. Scatter plot matrix charts can be used to visualize location data with varying properties. These charts are also called dot plot maps [10]. The visual variable used is often only positional, but different colors can be used to represent different types of data. An example of this data type is seen in figure 2.5.

Choropleth maps are used to color geographic units, such as countries or municipalities, based on quantitative values. These maps differ from heatmaps by having distinct borders between areas of interest. The choropleth maps provide nice visuals, but they come at a drawback, because data populations are rarely evenly distributed causing distortion in the visualizations. The borders can either be drawn as lines to emphasize the borders or be colored the same as the filling which would emphasize similar adjacent data points. The visual variables are position and different color saturations or lightnesses. An example of this chart can be seen in figure 2.6. [10]

Bubble plot charts draw different sized circles on a map to represent different magnitudes. The biggest strife of this type of chart is its tendency to get cluttered. Often the circles

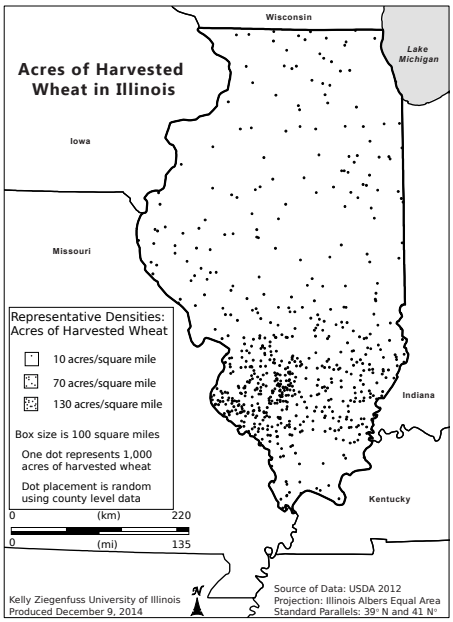

Figure 2.5. *Scatter plot matrix visualization of acres of wheat harvested in Illinois from [9].*

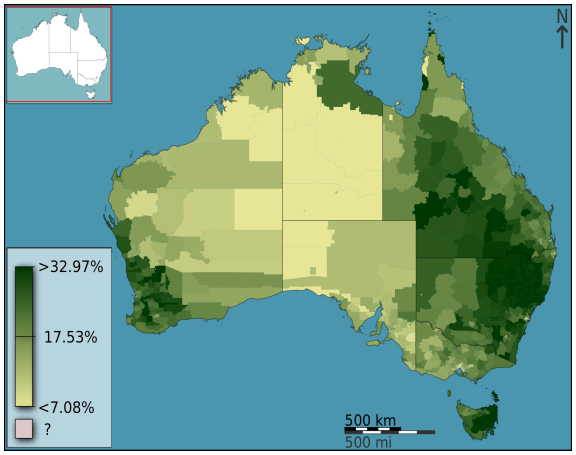

Figure 2.6. *Choropleth of people who identify as Anglican as a fraction of total persons, in Australia, Australia, according to the 2011 census results from [9].*

might start to overlap causing confusion for the viewer. Different colored bubbles can be used to represent different varieties of data. The visual variables are position, area and different colors. An example of this chart can be seen in figure 2.7. [10]

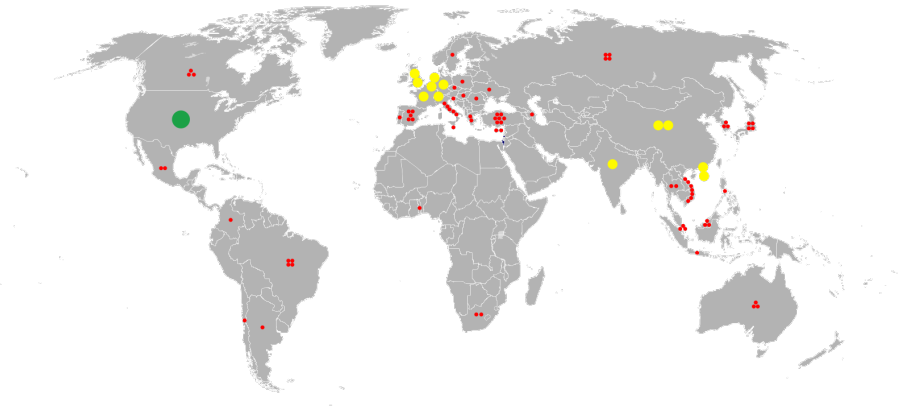

Figure 2.7. *Bubble map of Israeli exports, based on 2016 data from [13].*

2.7 Business development

It is thought that any given business process works in such a way that, first information is collected, then manipulated and then the results are used to make business decisions [11]. Decision vary from significant decisions that drive the course of strategies within the business, to decisions that are more narrow in nature that do not significantly affect the business, such as optimal advertising. Anything ranging between these two can be decision on how to package your product more cost-effectively, that are important decision, but less critical in nature. [11]

Businesses usually have some metrics to measure their performance. Of course, in a perfect world every business decision made would be the perfect one to lead to optimal business performance. Optimal decisions are however often difficult to make, because of the lack of insight on the business process. [11] These metrics can be turned into a Business Intelligence process.

2.7.1 Business intelligence

Business intelligence is defined to be the set of techniques used to convert raw data into information, which can be then used to inform decision-makers [11] [8]. The techniques can give decision-makers a view of the past, the present, and the future in the form of predictions. Business intelligence tools can range from standard reporting tools to data warehouses, data mining and online analytical processing. [8] This master's thesis visualization pipeline was used to create a standard utilizing tool, but could eventually be transformed to be a part of a data warehouse.

In businesses, additional information does not always provide more insight to the problems, as the plethora of data can often overwhelm the organizational deciders. This term is often

called ”analysis paralysis”, which is defined as the impulse to postpone decision-making until more data is available to simplify the decision. [11]

The paralysis can be abated by filtering the data, and perhaps by giving the correct data to the correct people. The decision-makers rarely need all the data at once and are usually more inclined to make a good decision when given the specific information regarding the decision at hand. [11]

When it comes to giving the right data to the right people, upper management often wants to see an overview of the data to make decisions that drive the strategies for better long-term value generation. Operational decision-makers get more value in their daily activities by getting information regarding the optimization of their processes. Optimally, this information could be implemented into the process itself by streaming the analysis reports. This would track performance of day-to-day activities and indicate when better decisions are being made. [11]

2.7.2 Business Intelligence assessment models

Different types of Business Intelligence assessment models exist to measure the maturity of an organizations business intelligence [14]. These methods range from technical assessments to more process centric assessments. Some of these assessments go through the steps a business can take from having no analytical capabilities to be a serious analytical competitor, while others focus on assessing how much more mature the business intelligence needs to get to reach business goals. [14]

Most of the Business Intelligence assessment methods require an already functioning Business Intelligence to be able to assess anything. In the case of Business Intelligence Maturity Hierarchy, we are able to assess the Business Intelligence from a technical point of view at a very general level. It is to be noted, that the assessment is regarded as incomplete in terms of a complete Business Intelligence assessment, but in the case of this master’s thesis we can still utilize it, because we are only forming a baseline. [17]

The Business Intelligence Maturity Hierarchy consists of four stages. The first step is the data stage, where the organization gathers, filters and sets a standardization for the data models, keeping the data set consistent. The goal in this step is to have a good basis for a future Business Intelligence process, with a high-grade data set which is standardized and clean. [17]

The second step is the information stage. In this stage the data is given meaning through dashboards and graphs. The key factors for company performance should be identified at this stage within the company. [17]

The third step is the knowledge stage. In this step the business should already be utilizing the data, for example through what-if analyses. Finding the different trends present in the data should become familiar in this phase to identify the origin of business problems. [17]

The final step is the wisdom stage, where business decisions should be able to be made well in advance based on the information gathered from the Business Intelligence. In this stage the organization should have a clear advantage against competitors, because of the data. [17]

3. THE VISUALIZATION PIPELINE DESIGN

The following chapter goes through the process of choosing the different visualization styles, technologies, approaches, and the pipeline's Business Intelligence assessment method. The chapter utilizes the researched theoretical background as well as the studied assessment methods from section 2.

3.1 Visualization chart selections

The data visualization charts of section 2.6.3 are gone through in this section, and the final decisions are explained in detail.

In the master's thesis we were presented with data that had a positional value as well as other properties, the taxi ride state being the most important.

The choropleth map could almost immediately be excluded for two reasons. The first being the fact that, it is extremely difficult to find a detailed choropleth map of the Tampere area. The second being its disadvantages when it comes to presenting locations accurately, which in some analytical cases can be quite critical, e.g. researching how crowded taxi poles are.

Next we could exclude the bubble map, because the data we are analysing has nothing to do with magnitudes. The only data type that varies in magnitude is the speed of the car, which in itself does not bring much analytical value. It can almost be argued that the bubble map without the level of magnitude would be the same thing as a scatter plot matrix.

The two chart types left are the scatter plot matrix and heatmap. The heatmap provides very broad visualization capabilities due to its high amount of visualization variables. Scatter plot matrices are basically colorable markers, which suffice in showing the taxi locations along with some property by utilizing the change in color.

The visualization types best suited for portraying location-based data were either heatmaps or scatter plot matrices, the preference being on the heatmaps, as they offer more options in the name of configurability.

The optimal choice would be a heatmap with an underlying scatter plot matrix, to help visualize single taxi locations, instead of only a cloudlike representation of multiple taxi locations.

The final choice for each approach is based on how well the selected tools support these two different visualization types. If a tool supports neither of the aforementioned visualization types, they are deemed unfit for the visualization pipeline in section 3.2.1. If the tool supports both, then both are used in unison.

3.2 Technology choices

The technology choices for the pipeline were made using the theory section to guide to good decisions. The basic concepts of the MVC layered web architecture were applied to the web application section of the pipeline.

The visualization pipeline was designed using the first approach described in 2.1, even though it is considered a conventional method for data pipelines. The disadvantages described in the Journal of Big Data do not apply, as the raw data was not wished to be kept in full. On top of this, the required pipeline architecture is simple enough for the disadvantage of replaceability to be an insufficient drawback.

The only transformation done within the pipeline was the filtering of unnecessary and repeating properties. The lesson learned from section 2.1 was to move analytics calculations into the front end of the pipeline, as it significantly improved the pipeline efficiency. The theory gathered from remote rendering also support this architecture model and provides value for end users who want to view the analytics on mobile or lower end machines.

3.2.1 Visualization tool selection

For the thesis, a significant number of visualization tools were studied for comparison. The tools were searched from various sources a typical software developer might find tools from. Most of the tools were found from popular search engines using search terms such as ”Web visualization tools for data”, ”Visualization tools for data”, ”Location based data visualization tools”. On top of this, word of mouth between company co-workers and thesis instructors was used to get a feel for possible interesting modern tools to compare.

First the tools were divided into three categories, tools with built-in geographic visualization tools, tools with geographic plugins, and tools with no built-in or plugin-based solution for geographic data. The last usually included some sort of scatter plot chart, where you could basically implement a static image of a map.

After the initial categorization, the tools were tagged with properties that defined whether or not they included monetization, were open-source or otherwise completely free. Some tools included free tiers and were tagged as both paid and free. Any other relevant tool information was added as a separate comment. The search resulted in table 3.1.

The search results from table 3.1 are narrowed down further. The tools are narrowed down to three different approach styles that are explained in the following section.

3.3 Approach selection

In the previous section, we narrowed down tools based on their monetization standards and based on their built-in functionalities. In this section we further narrow down the

Table 3.1. *Visualization tools from initial search results.*

	Cost	Additional information
Built-in geographic data visualization		
Amcharts	Paid	
D3JS	Free	
Datawrapper	Free / Paid	
FusionCharts	Paid	
Google Charts	Free	
Highsoft Highmaps JS	Paid	
Infogram	Free / Paid	
Mapbox GL JS	Free / Paid	
Plotly	Free / Open-source	
Polymaps	Free	Latest release in 2011
Tableau	Paid	
Geographic data visualization plugins		
Grafana	Free / Open-source	
Leafletjs	Free / Open Source	
No built-in geographic visualization		
Chartblocks	Free / Paid	
Chartist.JS	Free / Open-source	
Charts.JS	Free / Open-source	
Rawgraphs	Free / Open-source	
Sigmajs	Free / Open-source	

tools into three fitting categories based on the different audience types we categorized in section 2.6.1. These categories were organizational decision-makers, analysts and activists. These three are put into a table based on their different features stated in section 2.6.1 in table 3.2. In table 3.2 the different features are compared to each audience type. A ”+”-symbol is marked when the audience type is known to achieve benefit from the said feature. A ”-”-symbol is marked if the feature is generally known not to benefit from the feature. Nothing is marked if the feature is not explicitly good or bad, as some features are more relative to the context it is presented in.

Table 3.2. *Approach selection table.*

	Organizational decision-makers	Analysts	Activists
Simple visualizations	+		+
Interacting with the data		+	-
Open-ended visualizations	-	+	-
No self-interpretation of data	-	-	+
Data storytelling	+		+
Data art		-	+
Technical tools		+	-

Based on these facts we are able to create three different types of approaches with each audience type in mind. These approaches are referred in the master’s thesis as ”tailored”, ”lightweight” and ”technical”. The different approaches are used to generalize which kinds of audiences could possibly use each framework in future works. On top of this, these approaches bring valuable insight on different technical or qualitative problems one might face implementing the pipeline with the different approaches.

The three following sections go through the technologies chosen for each approach.

3.3.1 Technical

The first approach was to use a more ”technical” approach with a notebook-type platform to run the visualization tools, such as Jupyter Notebook, Apache Zeppelin or Grafana along with a visualization tool such as Plotly or D3JS.

For the master's thesis, Plotly was chosen as the visualization tool and Jupyter Notebook was chosen as the platform for the ”technical” approach. The choice was made due to the fact that Jupyter Notebook supports Plotly in great extent, and the choice fits the audience type very well.

In this case, the client would need to be technically inclined, due to how technical the notebook type approach is. The client would in this case be a analyst, or a technically inclined organizational decision-maker. A solid understanding of programming is almost a mandatory requirement, especially if the client wishes to customize the visuals themselves.

3.3.2 Lightweight

The second tested approach was to create a ”lightweight” solution. This would technology-wise be a very bare-boned solution, but still capable of making impactful visualizations.

In this master's thesis, Mapbox GL JS was chosen as the visualization tool and a simple Brackets server was chosen as the platform. The choice was made, because of Mapbox GL JS' extensive API for visualization, thus allowing it to run in a single file while making impactful visualizations.

The end result would not require technical skills and would provide pretty straightforward analysis of the subject. The audience would either be the activists or less technically inclined organizational decision makers.

3.3.3 Tailored

The last approach was to develop a ”tailored” and technical solution using a modern web framework, for example Vue.js or Angular 2. The biggest difference between the ”lightweight” solution and ”tailored” solution is being more demanding on the hardware as well as more demanding for the developer and having more options in terms of customizability.

In this master's thesis, Google Charts was chosen as the visualization tool and Vue.js as the web framework. The selection was mainly made due to Vue.js' extensive web functionalities and the possibility to import Google Charts as a node package with ease.

With this approach a highly interactive dashboard would be the end result. The audience would be the analysts or the organizational decision makers.

3.4 Programming languages

The selection of the back-end programming language was quite straightforward. The requirements for the visualization back end was to be able to handle concurrency, while still retaining the capability to run it on minimal hardware. In addition to this, the back end required REST capabilities.

The visualization back end was developed using Vert.x, because it is event driven and non-blocking, allowing the back end to be run on minimal hardware while maintaining asynchronous functionality. Vert.x supports equally a multitude of programming languages, and Kotlin was selected for the thesis as it was the most familiar. [23]

The front end did not have requirements concerning the programming languages, thus the programming languages selected for the front end were chosen based on the primary programming language used in the documentation. This is enabled quicker development for the visualizations, because example code could be used as a baseline. The ”technical” solution used Python 3, while the ”tailored” and ”lightweight” solutions used JavaScript.

3.5 Visualization pipeline assessment method

For the master's thesis, one of the criterion was to provide value to clients via a visualization pipeline. As stated in section 2.7.1, there are many methods to assess the maturity of a Business Intelligence process, which the visualization pipeline aims to be. We will use Innogiant's client as a case study to evaluate if the pipeline provides value in general, as well as to evaluate how well the audience categorization works in practice.

In this master's thesis used Business Intelligence Maturity Hierarchy as the chosen method for assessing the visualization pipeline in general. The Business Intelligence Maturity Hierarchy consists of four different stages, and we used the first and second stage as a target objective for the visualization pipeline. Although the first two stages were set as the goals, the pipeline was compared to all of the steps.

The reason for setting the objective only on the first to stages was because they were clearly the most technical. The later stages were more focused on the organization culture and the adaptation of the Business Intelligence. These properties were outside of the scope of the master's thesis research question, as well as out of the scope of knowledge, as the visualization pipeline had yet to be deployed into use.

The first stage of the Business Intelligence Maturity Hierarchy is the data stage, where the organization has set up the basis for a working Business Intelligence. The first stage was described to collect, cleanse and standardize the received data. The goal was to be a starting point for introducing higher level Business Intelligence, by establishing an integrated, clean and high-quality data set [17]. The second stage was the actual utilization of this high-quality data by creating dashboards and graphs [17]. These were the objectives the

data visualization pipeline had to fulfil, apart from the performance indicators of stage two as previously mentioned.

4. IMPLEMENTATION

The following implementations were created based on the technology choices made in chapter 3. This chapter goes through the implementations and explains the choices made, as well as goes through obstacles faced when developing the implementation.

4.1 Pipeline architecture

The pipeline architecture follows an MVC design. Starting from the top layer, we have the data layer, which logs the taxi API data using the credentials stored in the Redis data structure. If the credentials are not available, the logger requests a bearer token from the taxi API and stores it into the Redis data structure for future use. The logger passes the data to the filter, which in turn refines and filters the data into a suitable form.

The service layer consists of the formatter, which formats the data into a GeoJSON or JSON object for the REST API, which in turn offers an endpoint for the web layer. The RESTful API provides two endpoints for the different data models.

The web layer consists of the HTTP server, where the user interface layer can access the data passed from the back end. The web layer also passes the requested time frame the visualizations are wanted from to the back end.

The user interface layer contains the visualization tools and provides a possibility for users to interact with the data and set the time frame from an input box or date selector for example.

This architecture is visualized in figure 4.1.

4.1.1 Data flow

The architecture of the visualization pipeline is simple. The taxis send location data to the taxi API back end, which in turn provides an endpoint for the visualization back end. The visualization back end filters and stores the data and provides the visualization front end with multiple endpoints depending on the data format that was required by the visualization tool. Some tools were able to take greater advantages of GeoJSON data compared to non-standardized data formats. This was the case of Mapbox GL JS.

In a real world implementation, the most likely scenario would be to only have a single REST API endpoint for the data, as there would presumably only be a single visualization tool, and thus a single data format.

The data flow is visualized in a flowchart in figure 4.2.

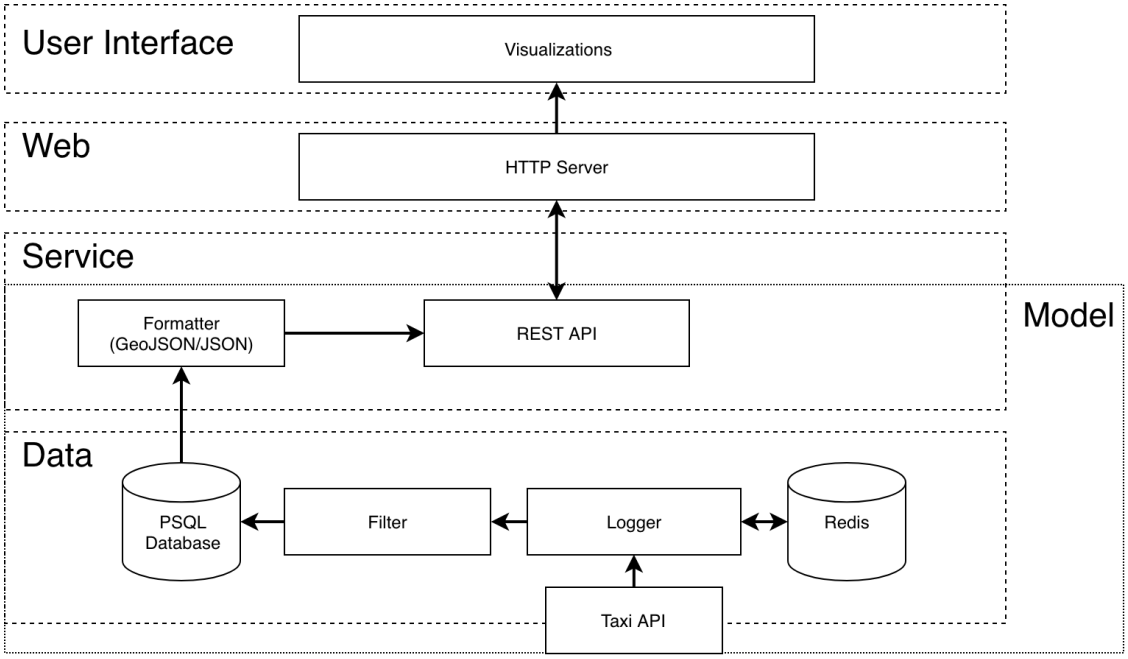

Figure 4.1. *MVC representation of the pipeline architecture.*

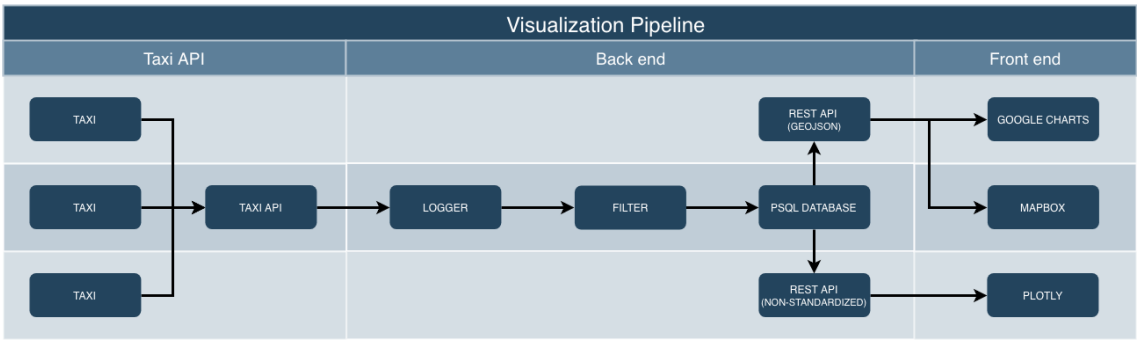

Figure 4.2. *Pipeline data flowchart.*

4.2 Back end

The back end utilized an API provided by the taxi company and was out of the scope of control. The back end consisted of a PostgreSQL database, a Redis data structure, and a Vertx server which uses Kotlin as the programming language. The back end utilized Gradle for build automation.

4.2.1 Data logger and filter

The data logging was implemented using a Vert.x worker verticle, the worker called a function based on configured values. The function authorized itself to the taxi location API via a bearer token, then it sent a request for the car locations.

The data logger at default settings logged the data from the taxi location back end every 30 seconds, as well as the taxi static data every 24 hours. The static data includes data such as the car model, car capacity and the taxi number. The static data was updated because it is not uncommon for the taxi numbers to change every once in a while. This was implemented to reduce the need for gathering of the same data multiple times. The two different calls were running inside the same worker verticle with different timer configurations.

In addition to the static data class, each time a location snapshot was taken, an identifier was stored alongside its creation time as a reference point for the snapshot. This was done to group each single call of the taxi location back end, which could later used to index the timeline that the front end would receive. The static data logic is visualized in figure 4.3.

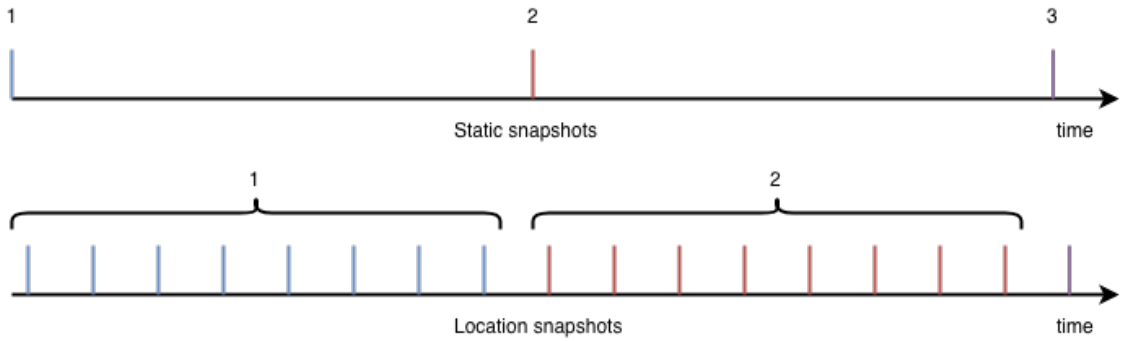

Figure 4.3. *Taxi location and static snapshot visualization.*

The taxi location API returned 167 lines of attributes per taxi for around a thousand taxis. A big part of the returned data was either out of date, repeated or otherwise useless for the data visualization, and this was why a lot of the data needed to be filtered. The JSON properties that were used in the visualizations for each taxi are represented in table 4.1.

4.2.2 RESTful API

The RESTful API was created using Vert.x handlers. In the event of a RESTful GET call to the back end, the back end first checked if the call was authorized before starting to retrieve

Table 4.1. *The relevant JSON properties provided by the taxi location back end.*

JSON property	Explanation
carID	The taxi's unique ID number
taxiNumber	The taxi's roof number
vehicleGroup	The car's model group key
location	Latitude and longitude of the taxi's location
speed	How fast the car is going
heading	Which way the car is heading
rideState	The taxi's state key
logTime	When the variable data was recorded in the taxi
snapshotTimestamp	When the snapshot was taken

the called resource.

Parameters for the start time and end time were required for the GET call, due of the sheer amount of data that would need to be passed if everything was returned. The data was filtered using SQL commands by getting only the rows of data which were created between the specified time as well as include a timestamp from within that time range. This is because the timestamp is provided by the taxis themselves and can contain outdated data that easily skews the data visualization.

The static taxi data of the specified time frame was combined with the location data via their identifying tags. The RESTful API returned a GeoJSON representation of the data if requested, otherwise it returned a normal JSON representation of the car snapshot data. The GeoJSON was simply mapped from the different taxi objects into a GeoJSON feature collection as GeoJSON points. The UML (Unified Modeling Language) class representation is visualized in figure 4.4

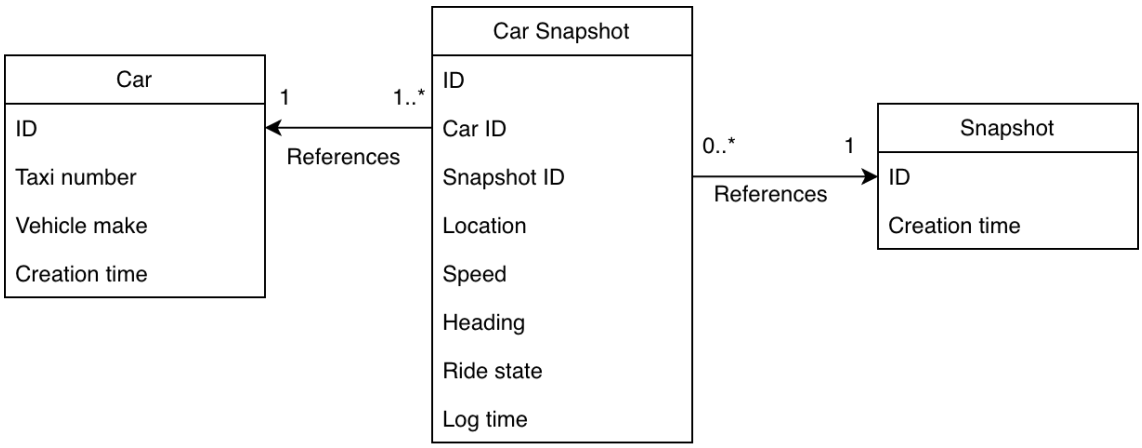

Figure 4.4. *UML representation of the application classes.*

4.3 Visualization implementations

Applying section 2.6.2 to the thesis, taxis that are not carrying passengers should be the easiest to spot, because it is important for the viewer to be able to pinpoint locations that

are not as successful as others. To achieve this the taxis not carrying passengers should use a brighter color than the taxis with passengers. The color chosen for cars with passengers was myrtle green, and the color for taxis without a passenger was a dark red. The colors were chosen with the ColorBrewer tool as reference using the qualitative scheme. Due to the fact that the quantity of taxis within a certain area wanted to be represented, a color range from ColorBrewer's sequential scheme was used as inspiration when choosing the gradient for the heatmaps.

In this section the three different implementations of the three approaches are explained in further detail. The three approaches are divided into the three different tools that were used. The Plotly section describes the ”technical” approach, the Mapbox GL JS section describes the ”lightweight” approach, while the ”Google Charts” approach describes the ”tailored” approach.

4.3.1 Plotly

The ”technical” approach used Plotly to visualize the data. Plotly was run on a Jupyter Notebook locally, using Python 3 as the programming language. Plotly does not provide a heatmap layer, instead it offers a scatter plot matrix on top of a Mapbox map. The tool requires each element to be put into arrays for each respecting category. This also requires a mapping function to go through the GeoJSON data, which causes additional strain on the pipeline and is not very scalable. A screenshot of this implementation can be seen in figure 4.8

Jupyter Notebook was highly versatile as it provided a whole different ecosystem than your own machine. In terms of Plotly, a heatmap layer would probably have been better a better visualization method, but the scatter plot matrix sufficed. Plotly could also produce static HTML files while in offline mode, and more versatile functionalities were behind a paywall.

4.3.2 Mapbox GL JS

The ”lightweight” approach used Mapbox GL JS for visualization. Mapbox GL JS, as the name indicates, is a Javascript library that utilizes WebGL to render interactive maps. Mapbox GL JS was run locally on a Brackets server, with Javascript as the programming language. Mapbox GL JS supports GeoJSON out of the box for its heatmap layer, and provides the developer with options to manipulate the heatmap based on the properties stored in the GeoJSON. For example adding more weight to a certain point based on a numeric value such as speed or time. Figure 4.5 shows a screenshot of the implementation.

Map Box GL JS also supports the addition of a scatter plot matrix layer on the map. On top of this, it has a very useful zoom functionality, where the heatmap fades away when zooming in and the scatter plot matrix can be configured to do the opposite. This functionality can be seen in 4.6, where the zoom level is set right where the heatmap is about to fade away.

The approach was completely run from a single file. This caused any additional functionality to clutter up the file. Some elements could be added on top of the map to control the timeline. Mapbox GL JS offered a lot in terms of functionality within a single file.

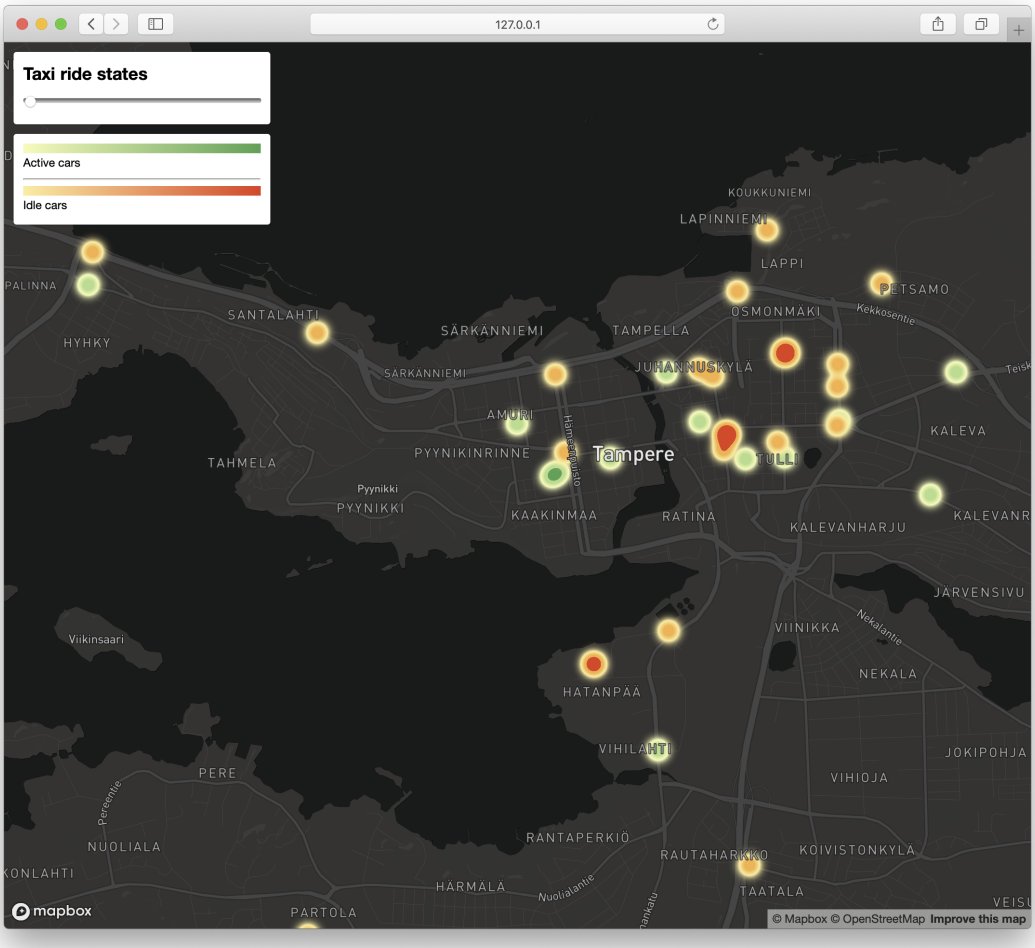

Figure 4.5. *Screenshot of the ”lightweight” implementation.*

4.3.3 Google Charts

The ”tailored” approach used Google Charts as the visualization tool. Google Charts was run in a Vue.js development server, using Javascript as the programming language. Google Charts offers a possibility to add heatmap layers on top of the map.

Google Charts gives the option to read GeoJSON files as a built-in functionality, but unfortunately the heatmap layer cannot utilize the GeoJSON data straight as is. The heatmap layer requires the data to be mapped into an array or Google’s preferred MVCArray type. The MVCArray offers more options in terms of data that updates. Requirements for extra mapping of data causes extra load on the whole process, and it would be wiser to not use the GeoJSON format in this case. The screenshot in figure 4.7 shows how the implementation looked.

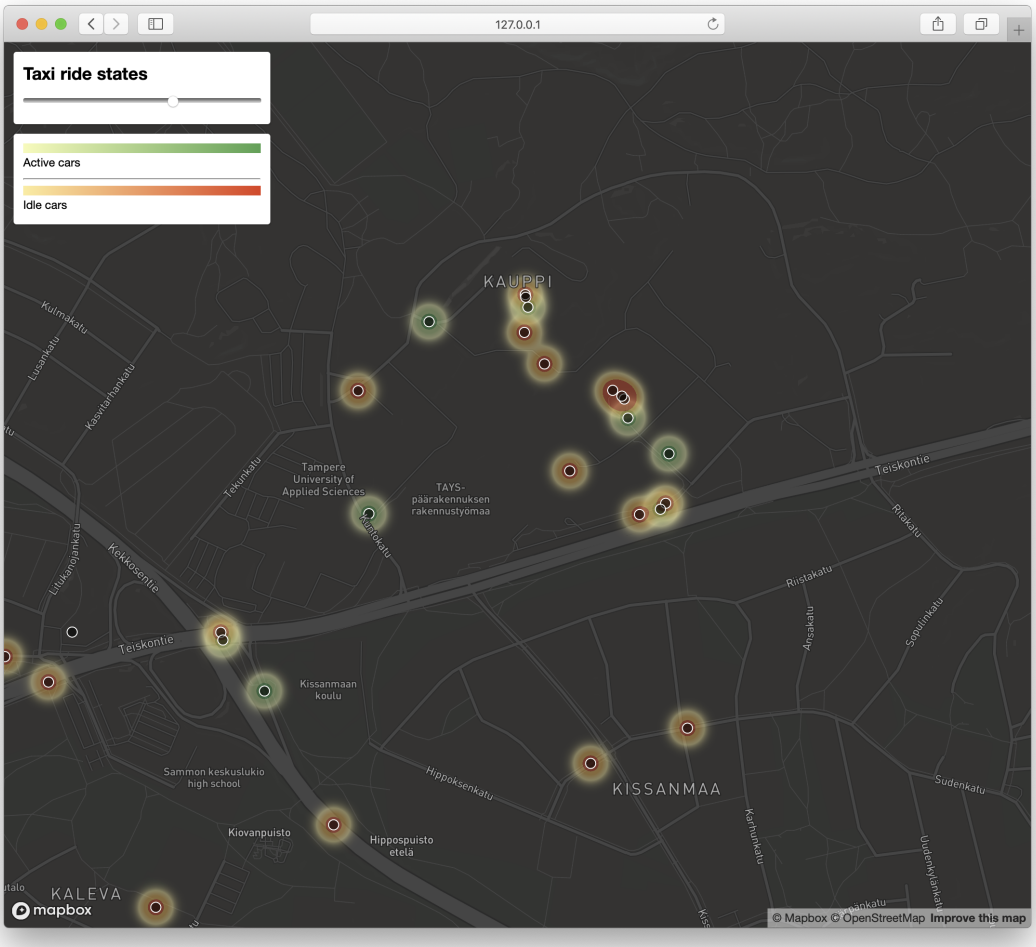

Figure 4.6. *Screenshot of the "lightweight" implementation's scatter plot matrix functionality.*

Using Vue.js offers great ways to expand the visualization into a dashboard. Sliders could be added to the site to control the timeline and data from the back end could be easily utilized elsewhere in the application.

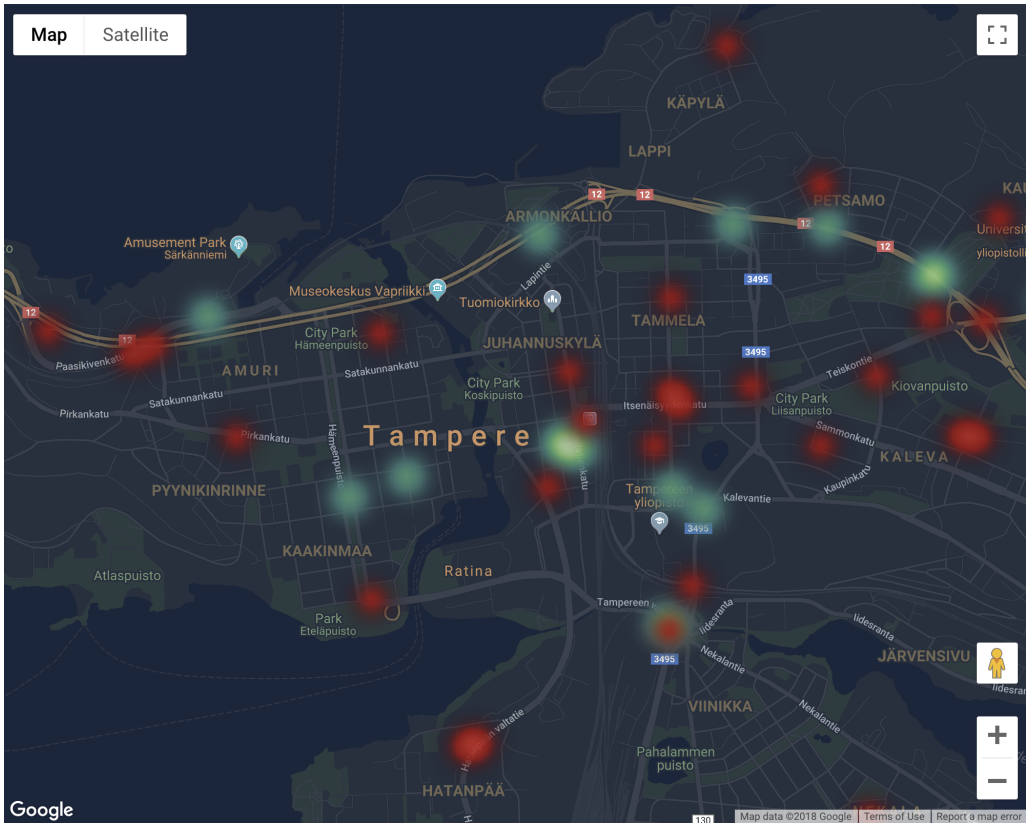

Figure 4.7. *Screenshot of the ”tailored” implementation.*

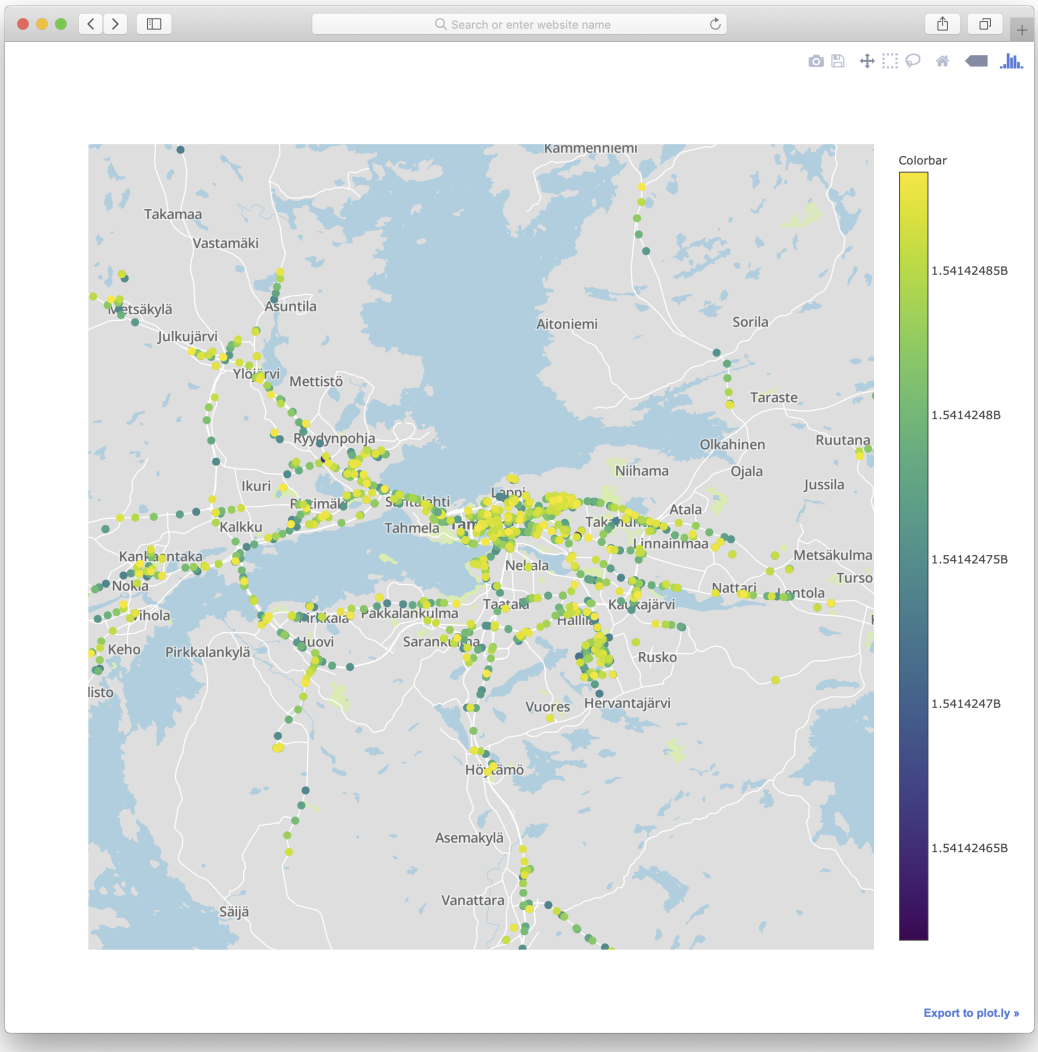

Figure 4.8. *Screenshot of the "technical" implementation.*

5. EVALUATION

This chapter evaluates the knowledge gained in chapter 4. The aim of this chapter is to evaluate the different approaches by using the case customer. First, the case customer's audience type is evaluated. Then the case customer is compared to the different approach designs for verification. In addition to this, the case customer is compared to a strictly technical overview of the approaches to evaluate which tool would fit the case customer best. Finally, the visualization pipeline is evaluated as a whole using the assessment method researched in 2.7.2 and by using the case customer as an example.

5.1 Case customer evaluation

Innogiant Oy's client, the case customer in this master's thesis, is a taxi company, which is completely owned by over 2000 shareholder taxi drivers. The company is responsible for taxi relay services as well as driving agreements. Annually the company drives over two million taxi orders in Tampere and its neighbouring municipalities. The company consists of an upper management, who use a dashboard to analyse taxi orders collected from the taxi mobile application.

The different audience types were defined in section 2.6.1. The three types of audiences were organizational decision makers, analysts and activists. Organizational decision makers were a very business focused audience who used the data to primarily make business decisions. Analysts were highly technical and liked to explore the data themselves. Activists were an audience that wanted to get some easy to grasp point from the data.

The visualization pipeline was built for the upper management in mind, thus it is a pretty clear case of the audience being organizational decision makers. This is because, the upper management makes decisions to improve their business based on the analytics the visualizations bring. It is pretty easy to rule out that the upper management are not activists nor analysts. They are not trying to prove a point beyond business decisions, which rules out the activist's category. They also do not come from a highly technical background which rules out the analysts as an audience category.

5.2 Approach comparison

Based on 5.1, the expected best approach would be the ”tailored” option, but to test the method of categorizing audiences used in this master's thesis, we compare the different approaches taken to the needs of the client. This is done to determine which of the approaches fit the client best based on the approach properties alone. The approaches are

compared to the problems defined in section 1.1. These problems were simply: does the approach create a clear visual representation of the data, does the approach bring value to the client.

In terms of forming a clear visual representation of location-based data, all of the approaches succeeded. The heatmap implementations of the ”tailored” and ”lightweight” implementations proved to be very informative and sparked inspiration on further analytical approaches to be used. The ”technical” approach succeeded in showing taxi locations over a certain time period, but in the case of trying to visualize multiple taxis at a taxi pole, it succeeded less notably. However, it did manage to paint a beautiful visualization of the taxi locations, thus passing the test.

Bringing value to the clients is a very broad metric. In terms of this master's thesis it can be defined as being able to form any kind of analysis from the data visualization. As previously mentioned with the cases of the ”tailored” and ”lightweight” approaches, they both were particularly proficient in showing congestion near taxi poles, which in itself is a very important analysis. As for the ”technical” approach we can easily form an image of how the taxis are spread throughout the Tampere area. The taxi company offers its services to neighbouring municipalities, and an analysis of how well they can serve them can be formed from the visualization. This is also the case for the ”tailored” and ”lightweight” implementations, and thus all three bring value to the client.

All of the approaches solve the given problem as expected. It is now only a matter of finding the correct technical specifications for the client.

5.2.1 Technical comparison

The approaches are compared from a technical standpoint, which in this case is essentially a tool comparison. The technical comparison was done to find the best technical solution of the developed solutions for the case customer. This was also done to see whether the categorization and approach prove to be a match.

The different approaches were compared to different technical metrics. The first metric was whether the tool was embeddable onto an existing web application. All of the three visualizations can be embedded into a web application. The only problematic approach was the ”technical” approach, as Jupyter Notebook as a platform is less ideal for embedding into an existing dashboard. It can be argued that in this case comparing the framework is not valid, because it can be argued that you would not embed Vue.js development server or a Brackets server into another application, but due to Jupyter Notebook being such a integral part of the ”technical” approach, we found that it should be compared as a whole in terms of embeddability.

The next comparable metric was how easy is it to modify the graphs on the fly. This is where the ”technical” approach along with Jupyter Notebook starts to show prominence.

Being able to alter the source code of the visualizations allows for quick changes to the graphs on the fly. The ”tailored” and ”lightweight” approaches both lack in this sense, as the frameworks are not in the hands of the end users as closely as with Jupyter Notebook.

The third metric was how easy is it to expand on the visualization into a business intelligence dashboard. This metric is in the case if the client would not have a business intelligence dashboard to start with. The ”tailored” and ”technical” approaches both allow for a well-designed dashboard to be implemented around the solution. In the case of the ”lightweight” solution this is obviously not possible, as it was designed to be a standalone solution. You could argue that you could add functionalities to the ”lightweight” and eventually it would suffice as a whole dashboard, but due to its single file approach, it is considered bad practice and is thus not considered easily expandable.

The final metric was how much configuration is required for the setup of the visualization solution. This is where the ”lightweight” approach is superior compared to the other approaches. It is designed in such a way that you can get a working prototype with minimal effort. The ”tailored” and ”technical” approaches both require more setting up, because of the frameworks being more sophisticated. The ”technical” approach requires the most setting up as the Jupyter Notebook creates a fully new ecosystem for the visualization tools.

When the visualization tools are compared to the audiences they were categorized to, we found that it was particularly difficult to satisfy the activist audience. Simple designs are hard to produce with tools that utilize maps. If a tool would have been found that supports choropleth maps for Finnish municipalities, we could have achieved simpler designs and thus satisfied the activist audience better. Many of the tools offered choropleth maps of continents, e.g. Europe and Africa, or even a map of the United States, but for such precise location-based data these large-scale choropleth maps would not provide value.

The heatmap and scatter plot matrices were a good choice for presenting location-based data. They both allowed for locating single taxis. The scatter plot matrix did this better, but with enough zoom the heatmap also allowed this. The best option obviously was when both were implemented simultaneously. In addition to this, the heatmap also allowed taxi groupings to be visualized on the map as brightly colored spots. The groupings were easy to locate and when checked against a map, they were more often than not taxi poles with idle taxis.

The colors used in the implementation proved to be visually pleasing to the development team. With help of the ColorBrewer 2.0 tool, picking the colors proved to be a simple task once the different data types the tool offers support to were understood. With the use of the tool, the visualizations were made to provide the maximum amount of value to the client, due to harmonious colors. Later, by utilizing the tool the visualizations could be made to support color-blind modes.

When it comes to the case customer, with this material taken into account, we can easily rule out the technical choices of the ”technical” approach. The Jupyter Notebook is too

technical for the needs of the taxi company. We can also rule out the framework choice of the ”lightweight” approach. The minimal configuration requirements of the ”lightweight” approach do not outweigh the expandability options the ”tailored” approach offers with its modern web framework. However, the visualization tool used in the ”lightweight” approach offers a more efficient way to handle the data, which can in fact be used in the framework of the ”tailored” approach.

5.2.2 Case customer solution

The findings of 5.2 and 5.2.1 are combined in table 5.1.

Table 5.1. *Findings for the approach comparison and technical comparison.*

	”Tailored”	**”Lightweight”**	**”Technical”**
Creates clear visual representation of the data	x	x	x
Brings value to the client	x	x	x
Embeddable	x	x	
Easy to modify graphs			x
Easily expandable	x		x
Requires minimal configuration		x	

Based on the comparison of the three approaches, we have found that in the case of the case customer, a single approach of the three fills the requirements. This approach is the ”tailored” approach. The client requirements requested for an application that could be easily implemented alongside or within the current dashboard solution. The preference being on the latter. This would rule out the ”technical” approach, as the solution would not be as easy to embed within the current dashboard solution. This leaves only the ”lightweight” and ”tailored” solutions as viable options.

The findings for the ”tailored” solution’s visualization tool choices were not optimal in a technical sense. This is because Google Charts was not capable of manipulating GeoJSON data as easily as the ”lightweight” solution’s visualization tool – Mapbox GL JS.

The ”lightweight” solution was not very expandable with new features, and limited the development team technically, by design, nonetheless. With all things considered, the findings show that the ”tailored” approach would be the best approach for the case customer, especially if Mapbox GL JS would be implemented. These findings would also infer that the visualization tool would be pretty insignificant when it comes to the approaches. The frameworks would in this case be the biggest factor.

This finding shows promise for the audience-oriented approach for data visualization pipeline design. The case customer proved to get the most value from the designated approach for its audience group. Even though the visualization tool would have been more optimal with the ”lightweight” approaches technical choice, this is irrelevant, because it can be argued that the framework would be the more defining choice in this case.

5.3 Visualization pipeline evaluation

The metrics to measure the success of the pipeline were for the pipeline to comply with modern architecture design as well as to bring maximum value to clients. We used the case customer to assess the success of the pipeline by using the method from section 2.7.2.

When evaluating in technical terms, we can be quite satisfied with the results. The pipeline was designed by utilizing MVC web application layer design. The pipeline design enabled for individual parts of the architecture to be replaced without having issues with cross dependencies. This was exceedingly apparent in the front end side of the pipeline, because there were the three different approaches which each used different technologies for the visualization. However, this was also the case when it came to the back end, as it can also be replaced without having an affect on the front end.

The method mentioned in section 2.7.2 is the Business Intelligence Maturity Hierarchy, which is used to assess the maturity of an organizations Business Intelligence. The baseline before the master's thesis visualization pipeline for the case customer was that no taxi location data was being gathered nor analyzed. Business Intelligence was non-existent when it came to location-based taxi data. The goal was to reach the first two steps of the hierarchy to count the visualization pipeline as a success. The analysis of the requirements for all of the steps are in table 5.2.

***Table 5.2**. Business Intelligence Maturity Hierarchy requirements for the first two steps that are met and not met.*

Requirements	Met	Not met
Step 1: Data		
Data is gathered	x	
Data is filtered	x	
Data is standardized	x	
Step 2: Information		
Data is given meaning through graphs or dashboards	x	
Key performance factors recognized within organization		x
Step 3: Knowledge		
Data is utilized through different analytical methods		x
Trends can be found within the data		x
Step 4: Wisdom		
Decision can be made in advance because of Business Intelligence		x
Clearly ahead of competitors as a result of Business Intelligence		x

As seen from table 5.2, the requirements for the data step are fully met. The technical side of the information step is met to full extent, but the organizational side cannot be met at this point in time, as the focus of this master's thesis does not extend that far. This means that the knowledge step and the wisdom step are left out as well, due to them not fitting the scope of the master's thesis.

When assessing the success of the pipeline without a specified assessment model, we can use the fact that the visualization pipeline provided value to the case customer by giving

tools to analyse their core business, which in itself is also an indication of success in terms of gaining insight on their business.

The visualization pipeline was designed to utilize remote rendering of the data, in this case primarily the filtering and formatting of the data. This removes the redundancy of having to calculate the same things multiple times on the client's different machines, thus saving in compute times. Nevertheless, it is to be noted that the visualizations themselves are rendered on the client machines. This is due to the fact that none of the visualization tools used had the option to render the visualizations out of the client. It also allows for the visualizations to not be bottlenecked by the network.

Design decisions are non-compliant with big data, due to not using tools such as Apache Spark or Kafka. It is worth mentioning though, that the data volume, velocity nor variety were great enough to justify a data intensive tool to be used. The pipeline at no point showed signs of struggle with the data at hand. In the case of scalability, the thesis raises questions. If the visualization pipeline is wished to be used in real time, or with tens of thousands of taxis, it is unclear if the current pipeline developed for the master's thesis would suffice. Then again, due to the smart architecture design, the front-end would not need replacement, thus not requiring the whole pipeline to be replaced in such a case.

In terms of real-time support for the visualization pipeline, the visualization pipeline enabled each of the chosen frameworks to support displaying data in real-time. The ”technical” approach is the least optimal of all the approaches when comparing the real-time capabilities, because it created a static HTML document, which would need to be created again if the data would be updated.

The main limitation of the pipeline was the back-end's data layer, and how quickly the API could be called. Theoretically, the back end was capable of passing the data as soon as it was logged by the taxi, but it was unclear how often the taxis update their location. The front end was able to handle the incoming data traffic with ease, and the implementation of real time data should not require extensive development on top of what was already implemented in the pipeline.

5.4 Research use cases

The possible use cases gained from this master's thesis are diverse. Companies planning visualization pipeline designs for their clients can utilize the audience-oriented approaches used in this master's thesis as a starting point for designing the visualization pipeline. Evaluating the client audience category proved to be an effective way to find the angles to maximize the value provided.

The work also proves the functionality of a web application-based visualization pipeline, which can be used as a basis for future pipeline designs. They can also study the problems faced to avoid them themselves.

5.5 Research strengths and limitations

The research methods are evaluated in the section. The strengths and limitations of the master's thesis are discussed to ensure the transparency as well as the integrity of the research.

5.5.1 Strengths

The research strengths in this master's thesis are the broad use of different sources to validate the visualization pipeline's design and architecture. The visualizations as well as the visualization audiences are studied to a good extent to set the basis of the whole research. The use of a case customer gives the study solid information on how well the theory parallels with practice.

The master's thesis creates a good baseline for further study in creating an audience-oriented process for visualization pipeline design.

5.5.2 Limitations

The limitations of the study come from the possible errors made in evaluating the different approaches for each audience type. Other than the initial filtering, the tools were selected based on the different possible approaches, which in itself is not inexplicable. Use of different tools for each approach could cause variation in results when using a case customer, which is the reason why the tools themselves were tried to be left out of the equation when evaluating the correct solution.

As for the tools used for visualization, it became clear that for most cases it is irrelevant which tool is used in each approach, as often the different tools are interchangeable between the approaches. This was the case with the ”lightweight” tool providing more value in a technical sense, while the ”tailored” approach used a more sophisticated framework, and thus provided more value in that sense. When it came to location-based data, namely the ”lightweight” approach was hardest to implement in a simplistic fashion, because of location-based data's complex nature.

The categorical evaluation of the case customer was also done only using literary sources, which could have been extended into an interview or questionnaire to really extend the knowledge on the client.

It was also stated in [14] that the Business Intelligence Hierarchy model might not be the most valid assessment method for Business Intelligence, but it was chosen for this master's thesis regardless of that. This is due to the fact that a technical baseline objective was wanted. With a set objective, we were able to assess the success of the pipeline instead of just analysing the values it provided for the case customer.

The case study also only provided a narrow glance into actually proving the effectiveness of the audience-oriented approach. A lot more test cases, especially from the other audience groups, are required before the approach can have any proper ground scientifically.

6. CONCLUSION

This chapter defines a conclusion based on the evaluation made in chapter 5. First, the main points of the master's thesis are concluded. Finally, Possibilities for future work surrounding this topic are discussed.

6.1 Thesis conclusions

The findings in chapter 5 were valuable in coming to a conclusion. We were able to evaluate the client needs as well as to evaluate how well the visualization pipeline was designed and implemented.

6.1.1 Visualization approaches

The goal for the master's thesis was to provide generalized approaches to creating a visualization pipeline for different audience types. The audience types were organizational decision-makers, analysts and activists. Each of these audience types were designed a separate visualization front end using different visualization tools. The method was tested with a case customer.

In terms of the case customer, we found that the ”tailored” approach was the best fit for this client case. Not only did the client fit into the audience role of organizational decision-makers, but the approach provided the client with an embeddable solution into their existing system, while retaining great possibilities for expansion with new analyses. The case customer gained great value, and more specifically the most value from the exact visualization pipeline it was assigned. The case study showed signs of promise in terms of the possibility to utilize the audience-oriented approach in future visualization pipeline designs.

It was also noted that the visualization tools were deemed insignificant when comparing the approaches. This is because most of the tools are interchangeable between the different frameworks. The frameworks themselves were the defining factor between the different approaches, unlike first hypothesized where both the tool and framework would have a difference.

6.1.2 Visualization Pipeline

The visualization pipeline proved to be successfully engineered with the design used in this master's thesis. By using proper design methods, the different parts of the pipeline were

not strictly dependant on one another, which allows for parts of the pipeline to be replaced without having an effect on the data flow.

The visualization pipeline passed the set goals of the business intelligence assessment method in terms of the technical implementation. The organizational requirements were not met as they were not a part of the master's thesis study.

6.2 Future work

The research done with the visualization pipeline for location-based data creates many viable further research topics. In the front end of the problem, a lot can be researched when it comes to how the data is visualized. What kind of analyses could be done with the data at hand, how can the visualizations be improved further, what are the types of business decisions can you make from location-based data, are all good study questions.

From a technical point of view, the fact that the research does not compare different back end solutions leaves a lot of room for research within that domain. There are most likely more efficient ways to handle, filter and store the data in the back end. A good study question is also, when does data become big data? This could be used to specify whether a web application-based solution is sufficient or if a big data solution is needed.

From a data science point of view, it would also be very interesting to expand on the audience types. This master's thesis used very straightforward categories for the audience types for the sake of simplicity. In future works it would be interesting to define other niche audiences to design visualization pipelines for.

REFERENCES

[1] A. Bennett, Design Studies : Theory and Research in Graphic Design, Princeton Architectural Press, New York, 2006.

[2] N.A.C.D. Cathy Clerbaux, Carbon monoxide concentrations in spring, May, 2004. Available: https://upload.wikimedia.org/wikipedia/commons/f/f4/Carbon_Monoxide_concentrations_in_spring..jpg

[3] Color brewer 2.0 web tool, Axis Maps LLC, website, 2013. Available (accessed on 15.11.2018): http://colorbrewer2.org/

[4] Y.C.L.S.V.C. Deinum M., Serneels K., Web Application Architecture, Apress, Berkeley, CA, 2012, pp. 51–64.

[5] A.C. Doyle, Five Sherlock Holmes detective stories comprising: The red-headed league ; A case of identity ; A scandal in Bohemia ; That little square box ; John Barrington Cowles, J.S. Ogilvie, New York, 1895.

[6] S. Gillies, H. Butler, M. Daly, A. Doyle, T. Schaub, The geojson format, coordinates, Vol. 102, 2016, pp. 0–5.

[7] Json. Available (accessed on 19.11.2018): http://json.org/

[8] G. Keith, Business Intelligence, 2nd ed., BCS The Chartered Institute for IT, 2013.

[9] U.o.I. Kelly Ziegenfuss, Acres of harvested wheat in illinois, December, 2014. Available: https://upload.wikimedia.org/wikipedia/commons/3/38/Acres_of_Harvested_Wheat_in_Illinois_in_2012.pdf

[10] A. Kirk, Data Visualization: a successful design process, 1st ed., Packt Publishing, GB, 2012.

[11] D. Loshin, Business Intelligence, Morgan Kaufmann, 2012;2013;.

[12] A. McAfee, E. Brynjolfsson, Big data: the management revolution, Harvard business review, Vol. 90, Iss. 10, 2012, pp. 60–128.

[13] Y. Nasonov, Bubble map of israeli exports, based on 2016 data., May, 2018. Available: https://upload.wikimedia.org/wikipedia/commons/5/54/Israel_exports_bubble_map_2016.svg

[14] C.M. Olszak, Assessment of business intelligence maturity in the selected organizations, pp. 951–958.

[15] C. Pautasso, E. Wilde, R. Alarcon, REST: Advanced Research Topics and Practical Applications, 2014th ed., Vol. 9781461492993, Springer Verlag, New York, NY, 2014;2013;, 222 p.

[16] L. Pierson, Data Science For Dummies, 2nd;1; ed., For Dummies, US, 2015;2017;, 364 p.

[17] I.H. Rajteric, Overview of business intelligence maturity models, Management : Journal of Contemporary Management Issues, Vol. 15, Iss. 1, 2010, p. 47.

[18] T.M. Rhyne, Applying color theory to visualization, 2017.

[19] A. Sahay, Data Visualization, Volume I, 1st ed., Business Expert Press, 2016.

[20] S. Shi, C.H. Hsu, A survey of interactive remote rendering systems, ACM Computing Surveys (CSUR), Vol. 47, Iss. 4, 2015, pp. 1–29.

[21] U. Suthakar, L. Magnoni, D.R. Smith, A. Khan, J. Andreeva, An efficient strategy for the collection and storage of large volumes of data for computation, Journal of Big Data, Vol. 3, Iss. 1, 2016, pp. 1–17.

[22] Taxi market liberalisation set to alter fares and services in july, May, 2018. Available (accessed on 20.11.2018): https://yle.fi/uutiset/osasto/news/taxi_market_liberalisation_set_to_alter_fares_and_services_in_july/10192384

[23] Vert.x homepage. Available (accessed on 19.11.2018): https://vertx.io/